CONFIGURING CACHE MEMORY FROM A STORAGE CONTROLLER

BACKGROUND OF THE INVENTION

Field of the Invention

[0001] This invention generally relates to a storage controller that interfaces between host computer systems and a direct access storage device system. More specifically, the invention relates to deconfiguring cache memory from such a storage controller.

Background Art

[0002] In a large distributed computer system, a plurality of host computers and devices are typically connected to a number of direct access storage devices (DASDs) comprised of hard disk drives (HDDs). The DASDs may be organized in a redundant array of independent disks, i.e., a RAID array. A RAID array is comprised of multiple, independent disks organized into a large, high-performance logical disk. A controller stripes data across the multiple disks in the array and accesses the disks in parallel to achieve higher data transfer rates. However, utilizing multiple disks in an array increases the risk of failure. The solution in the art is to employ redundancy in the form of error-correcting codes to tolerate disk failures.

[0003] Not only is there is a risk associated with the failure of a hard disk drive in a DASD system such as a RAID array, but there is also a risk of failure at a point within a storage controller which controls read and write operations between host computers and the DASDs. The conventional storage controller is typically designed to handle hardware failure. One such storage control designed to handle certain hardware failures is the

clusters, each of which provides for selective connection between a host computer and a DASD. Each cluster has a cache and a non volatile storage unit (NVS). The cache buffers frequently used data. When a request is made to write data to a DASD attached to the storage controller, the storage controller may cache the data and delay writing the data to a DASD. Caching data can save time as writing operations involve time consuming mechanical operations. The cache and NVS in each cluster can intercommunicate, allowing for recovery and reconfiguration of the storage controller in the event that one of the memory elements is rendered unavailable. For instance, if one cluster and its cache fails, the NVS in the other cluster maintains a back-up of the cache in the failed cluster.

[0004] These dual-cluster storage controllers may also failover to a single node configuration if a memory upgrade is required. During these periods of failover, the system is running on a single node configuration, which is less fault tolerant than the normal dual node configuration.

SUMMARY OF THE INVENTION

[0005] An object of this invention is to improve storage controllers for interfacing between host computer systems and a direct access storage device system.

[0006] Another object of the invention is to maintain dual cluster fault tolerance in a dual cluster storage controller even during a memory fail or a memory upgrade.

[0007] A further object of the present invention is to deconfigure a cache memory, of a dual cluster storage controller, without failing over to the other cluster, thus maintaining dual cluster fault tolerance.

[0008] These and other objectives are attained with a storage controller, and a method of operating a storage controller, for interfacing between a plurality of host systems and a direct access storage devices system. The storage controller includes a first cluster including a first processor and a first cache, and a second cluster including a second processor and a second cache. The method comprise the step of directing data from the host systems through first and second data paths in the storage controller to the direct access storage system, wherein the first processor and the first cache are associated with the first data path, and the second processor and the second cache are associated with the second data path. Under a first set of defined conditions, the controller enters into a failover mode, wherein data directed to the first data path are routed to the second data path. Under a second set of defined conditions, the controller deconfigures the first cache without entering the failover mode.

[0009] The first cache includes a series of memory pages; and in a preferred embodiment, the deconfiguring step includes the steps of identifying selected pages of the first cache; marking each of said selected pages as unavailable; and after all of said selected pages are marked as unavailable, removing the selected pages from the first cache.

[0010] Further benefits and advantages of the invention will become apparent from a consideration of the following detailed description, given with reference to the

accompanying drawings, which specify and show preferred embodiments of the invention.

BRIEF DESCRIPTION OF THE DRAWINGS

[0011] Figure 1 illustrates a block diagram of a storage controller with which the present invention may be practiced.

[0012] Figure 2 schematically shows the control structure and memory configuration of one of the caches of the storage controller of Figure 1.

[0013] Figure 3 is a flow chart illustrating a preferred procedure for deconfiguring the cache memory.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0014] In the following description, reference is made to the accompanying drawings which form a part thereof, and in which is shown, by way of illustration, a preferred embodiment of the present invention. It is understood that other embodiments may be utilized and structural changes may be made without departing from the scope of the present invention.

Hardware Environment and Architecture of the Storage Controller

[0015] FIG. 1 illustrates a block diagram of the components and architecture of the preferred embodiment of a storage controller 2 which interfaces between host computers

or devices (not shown) and DASDs 46, 48, which include multiple RAID arrays. In preferred embodiments, the DASDs are magnetic storage units such as hard disk drives. The host computers and devices are connected to host adaptors 4, 6, 24, 26 via a bus interface (not shown), such as a SCSI bus interface. The host adaptors 4, 6, 24, 26 may be comprised of an Enterprise System Connection (ESCON) adaptor which provides access to ESCON channels and connections. Each host adaptor 4, 6, 24, 26 may be comprised of a series of host adaptors which connect to a host system.

[0016] In preferred embodiments, the storage controller 2 is divided into two clusters, cluster 0 and cluster 1. Cluster 0 is comprised of host adaptors 4, 6, a non-volatile storage unit (NVS) 8, a cache 10, a processor 12, a device adaptor bus 14, and device adaptors 16, 18, 20, 22. Cluster 1 is comprised of host adaptors 24, 26, an NVS 28, a cache 30, a processor 32, a device adaptor bus 34, and device adaptors 36, 38, 40, 42. A host adaptor bridge 44 interfaces the components of cluster 0 with cluster 1. The host adaptors 4, 6, 24, 26 are connected to the host adaptor bridge 44. In preferred embodiments, the bridge 44 is a dual master bus which may be controlled by one of the processors 12, 32 or one of the host adaptors 4, 6, 24, 26. In other embodiments, the host adaptor bridge 44 may include bridge technology to allow the bus to operate at its own clock speed and provide a buffer to buffer data transferred across the bridge 44. The bridge 44 interconnects the host adaptor 4, 6, 24, 26 with processors 12, 32. In preferred embodiments, the processors 12, 32 are symmetrical multi-processors, such as the IBM RS/6000 processor. Each processor 12, 32 maintains information on the configuration of the other cluster in order to re-route data transfers directed toward the other cluster.

[0017] The caches 10, 30 may be external to the processors 12, 32 or included in the processor 12, 32 complex. A processor 12, 32 in one cluster can communicate with the other processor, NVS 8, 28, and cache 10, 30 in the other cluster via the host adaptor

bridge 44. In preferred embodiments, the NVS 8, 28 is comprised of a random access electronic storage with a battery backup. The cache 10, 30, in comparison, is a volatile storage unit that cannot maintain data in the event of a power failure.

[0018] Device adaptor bus 14 interconnects the processor 12 with the device adaptors 16, 18, 20 22, and device adaptor bus 34 interconnects processor 32 with device adaptors 36, 38, 40, 42. The device adaptors 16, 18, 20 22, 36, 38, 40 42 interface between the storage controller and DASDs, or RAID array of hard disk drives. In preferred embodiments, the DASDs may be interconnected in a loop topology including multiple RAID arrays.

[0019] Because one device adaptor from each cluster 0, 1 is attached to each loop of DASDs, failure in one cluster and/or the device adaptors associated with the failed cluster will not prevent the functioning cluster from accessing the loop. Thus, no single point of failure in a cluster and/or in a device adaptor will prevent the other cluster from accessing a group of DASDs. Moreover, if a device adaptor, such as device adaptor 22, fails in a cluster that is otherwise functioning properly, then the re-routing to the other device adaptor 36 can occur at the device adaptor level. Alternatively, the failure of a device adaptor can be treated as a failure by the entire cluster, thereby transferring control over to the functioning cluster to access the DASD.

[0020] As mentioned above, when the storage controller runs in a single node configuration, the system is less fault tolerant than when the system is in the dual node configuration. It is, therefore, highly desirable to reduce the number of circumstances in which the system runs in the single node configuration.

[0021] In accordance with the present invention, a procedure is provided to deconfigure cache memory without failing over to the other cluster, thus maintaining dual cluster fault

tolerance. For purposes of example, this procedure will be explained with respect to cache 10. As will be understood by those of ordinary skill in the art, cache 30 may be constructed and operated in the same manner as cache 10.

[0022] With reference to Figure 2, the data in cache 10 is structured in a logical track fashion. For example, in an ESS implementation, one logical track 50 contains seventeen discontiguous 4K pages 52. There is a control structure 54 (referred to as a segment control block) for each 4K page and another control structure 56 (referred to as the directory control block) for each track.

Removal of Cache memory from the controller

[0023] Figure 3 shows a preferred procedure for removing cache memory from the controller. At step 60, the pages to be removed within a range are identified; and at step 62, a track id is stored in the segment control block 54 to identify which directory control block 56 pertains to the track containing the segment. At step 64, a determination is made as to whether the track is modified or unmodified. If the track is unmodified, then at step 66, the page is invalidated and marked unavailable. If the track is modified, then at step 70, a determination is made as to whether the track is pinned or bound.

[0024] If the track is modified but not pinned or bound, the track, at step 72, is grouped with other tracks for destage. Following a successful destage, the track, at step 74, becomes unmodified and the page is invalidated and marked unavailable. However, if the track is pinned or bound the memory management component proceed through steps 76, 80, 82 and 84. Specifically, at step 76, the management component allocates another free page or pages which are outside of the range; and at step 80, the track is marked busy. At step 82, the data is copied from the old pages to the new pages, and at step 84,

the old pages are marked unavailable. Once all the required pages are marked unavailable, the memory, at step 86, can be removed without failover to the other cluster.

[0025] While it is apparent that the invention herein disclosed is well calculated to fulfill the objects stated above, it will be appreciated that numerous modifications and embodiments may be devised by those skilled in the art, and it is intended that the appended claims cover all such modifications and embodiments as fall within the true spirit and scope of the present invention.